

.Debian

銀河系唯一のDebian専門誌

2014年08月23日

特集：Debian でタイルマップサービス



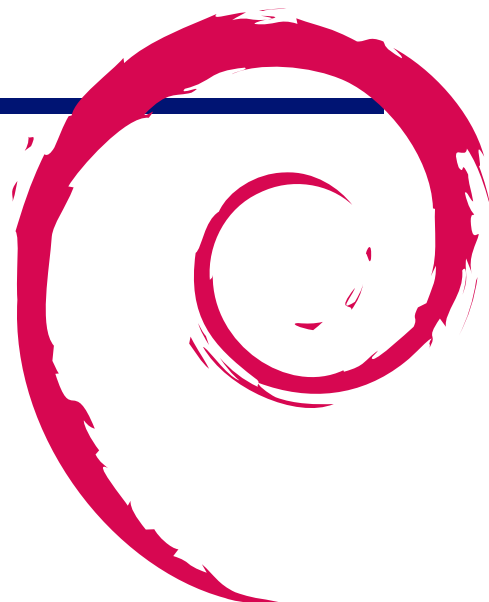
今日勉強会 の Debian トリビア

目次

1	事前課題	2	3	最近の Debian 関連のミーテ ィング報告	5
1.1	dictoss(杉本 典充)	2	3.1	第 115 回東京エリア Debian 勉強会 with 第 2 回 Debian パッケージング道場報告 . . .	5
1.2	吉野 (yy-y-ja.jp)	2	4	Debian でタイルマップサービ ス作ってみた	6
1.3	zinrai	2	4.1	タイルマップサービスとは .	6
1.4	田上 健太	2	4.2	タイルの生成	7
1.5	野島 貴英	2	4.3	タイルマップサーバの実装 .	8
1.6	wbcchsyn	2	4.4	まとめ	9
1.7	まえだこうへい	2	5	索引	10
1.8	野首 (@knok)	2			
2	Debian Trivia Quiz	3			

1 事前課題

野島 貴英



今回の事前課題は以下です:

1. 本日、何の作業をやるかを宣言ください。

この課題に対して提出いただいた内容は以下です。

1.1 dictoss(杉本 典充)

- 前回のパッケージング道場の続きをする
- VAIO P で linux-image-3.14 の場合に MP3 再生が倍速再生されるバグの調査

1.2 吉野 (yy-y-ja-jp)

DDTSS

1.3 zinrai

iodine,nikola を使ってみる。

<https://packages.debian.org/sid/iodine>

<https://packages.debian.org/sid/nikola>

1.4 田上 健太

新しく個人用ノート PC を買ったので、開発環境を構築していきます。

1.5 野島 貴英

xmris パッケージ化する。

1.6 wbcchsyn

6 月に発表したツールについて、頂いた意見の取り込みとフィードバックをおこないます。

1.7 まえだこうへい

<http://d.palmtb.net/pages/about.html#todo-from-maintainer-d> のバグ潰し

1.8 野首 (@knok)

- KAKASI のローマ字テーブル見直し
- Leap Motion, HID 送信機のハック
- ECS Liva の問題調査

2 Debian Trivia Quiz

野島 貴英



ところで、みなさん Debian 関連の話題においついていますか？ Debian 関連の話題はメーリングリストをよんでいると追跡できます。ただよんでいるだけでははりあいがないので、理解度のテストをします。特に一人だけでは意味がわからないところもあるかも知れません。みんなで一緒に読んでみましょう。

今回の出題範囲は `debian-devel-announce@lists.debian.org` や `debian-devel@lists.debian.org` に投稿された内容などからです。

問題 1. 2014/8/16 に Debian は誕生日を迎えました。さて今年で何歳？

- A 25
- B 21
- C 20

問題 2. 2014/7 時点の各国のアクティブな Debian Developer の数とそれぞれの国の人口との比率が最も多いのはどこの国？

- A Finland
- B Ireland
- C New Zealand

問題 3. OpenAmbit が 2014/7 にパッケージ化され sid へリリースされました。ところで何するパッケージ？

- A AmazonTV 用アプリ開発環境
- B Chromecast 用アプリ開発環境
- C SUNNTO AMBIT 用アプリ開発環境

問題 4. 2014/7/31 に technical committee により Debian のデフォルトの JPEG 圧縮伸長ライブラリとして採択されたのは以下のうちのどれ？

- A libjpeg6b
- B libjpeg8/9
- C libjpeg-turbo

問題 5. 2014/7/31 に technical committee から、「Debian パッケージは複数の init システムに対応する事」ということが再アナウンスされました。何のパッケージのドタバタがきっかけでしょう？

- A ftp
- B tftp-hpa
- C ncftp

問題 6. 2014/7/20 に squeeze (注: squeeze-lts ではない) の最後のアップデートが行われました。何回目のアップデートでしょうか？

- A 10
- B 9
- C 8

問題 7. 2014/7/31 に Jessie に搭載の可能性のある linux kernel のバージョンのアナウンスが行われました。バージョンはいくつでしょう？

- A 3.14
- B 3.12
- C 3.16

問題 8. VCS-*フィールドの VCS 情報を参照し、パッケージの changelog が合致しているかのチェックが行われるようになりました。以下のどのサイトで確認できる?

- A <http://qa.debian.org/cgi-bin/vcswatch>
- B <http://bugs.debian.org/>
- C <http://codesearch.debian.net/>

問題 9. パッケージに含まれているドキュメントに関して、lintian が新たな警告をするようになりました。以下のどれ?

- A HTML ファイル中の画像/CSS/JS/video リンクがローカルファイルを指していない場合、警告
- B HTML ファイル中の Debian の綴りが間違っている場合、警告
- C HTML ファイルが入っていない場合、警告

問題 10. 2014/7 に BTS の WEB サイトに reply のリンクがつくようになりました。このリンクをブラウザからクリックすると何が起きるようになった?

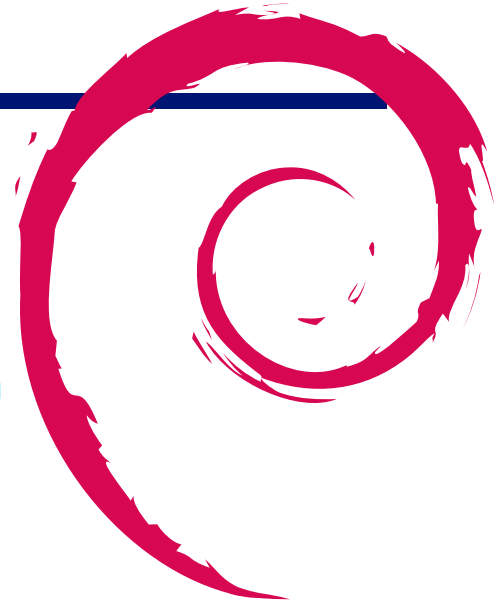
- A BTS 上で reply している次のメッセージが表示されるようになった。
- B BTS 上の reply 用のフォームがブラウザに表示されるようになった。
- C メーラが起動し、正しい Subject/宛先/引用が入るようになった。

問題 11. 2014/8/14 に Debian Installer Jessie Beta 1 がアナウンスされました。このバージョンのインストーラで導入される init システムは結局どうなった?

- A systemd になった
- B sysvinit になった
- C upstart になった

3 最近の Debian 関連のミーティング報告

野島 貴英



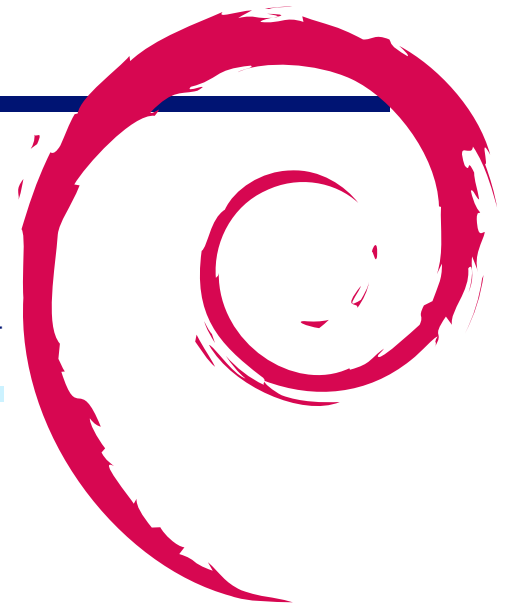
3.1 第 115 回東京エリア Debian 勉強会 with 第 2 回 Debian パッケージング道場報告

- (株) サイバーエージェントさんをお借りしての開催でした。途中参加の方も含め参加者 10 名。
- 10:00-16:00 までは第 2 回 Debian パッケージング道場を岩松さんにより進行、
- 16:00 以降はいつもの東京エリア Debian 勉強会の形式で、前田さんにより「Jenkins を用いたパッケージ作成の自動化」の発表

でした。

パッケージ作成については公式 Debian Developer の方による最近の事情も含めた説明が行われ、また、Jenkins を使いパッケージの生成を最後まで自動化されているのは素晴らしかったです。

最後に会場近くの飲み屋で宴会をしました。



4 Debian でタイルマップサービス作ってみた

なかおけいすけ

4.1 タイルマップサービスとは

タイルマップサービス (TMS: Tile Map Service) は、OSGeo 財団 (Open Source Geospatial Foundation) が策定した、地図をタイルとして提供するプロトコルです。タイルは、地図を領域とズームレベル毎に 256px × 256px の画像に分割したもので、図 1 のようにピラミッド構造になっています。大きな画像を小分割することで、必要な領域、ズームレベルだけを取得し、不要になった部分を開放することで、メモリの消費量を削減し、効率良く通信を行うことができます。タイルは、ズームレベルが一つ大きくなると、表示する範囲が 1/4 になります。例えば、ズームレベル 0 では、地球全体を 1 枚のタイルで表し、ズームレベル 1 では 4 枚、ズームレベル 2 では 16 枚... ズームレベル n では $2^n \times 2^n$ 枚になります。ズームレベル 16 になると、 $2^{32} = 4,294,967,296 \approx 4.3 \times 10^9$ 枚ものタイルが必要になります。

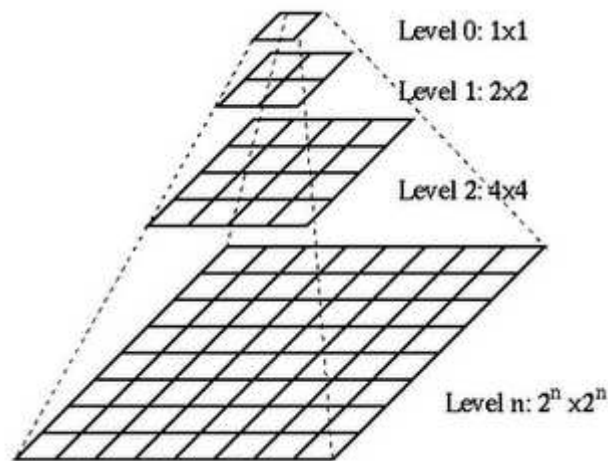


図 1 タイルのピラミッド構造 [1]

タイルは、HTTP で、`http://BASEURL/VERSION/TILENAME/z/x/y.png` という形式の URI で取得できます。現在、VERSION は 1.0.0 です。ここで z はズームレベル、 x 、 y は領域を表す整数です。 x 、 y は、 lat 、 lon を緯度、経度として以下のように書けます [2]。

$$x = \frac{2^z(lon + 180)}{360}$$

$$y = 2^{z-1} \left(1 - \frac{\ln(\tan(lat \frac{\pi}{180})) + \frac{1}{\cos(lat \frac{\pi}{180})}}{\pi} \right)$$

4.2 タイルの生成

タイルを生成するためには、表示するデータが必要です。今回は JAXA の水循環変動観測衛星「しずく」に搭載されている AMSR2 というセンサーで観測された海水表面温度データをタイルにしてみましょう。

「しずく」のデータは、登録が必要ですが一般公開されています [3]。登録をすると、sftp でデータをダウンロードできます。

```
$ sftp -oPort=2051 username@gcom-w1.jaxa.jp  
sftp> get AMSR2/YYYY/YYYY.MM/L3/SST_10/1/GW1AM2_YYYYMMDD_01D_EQOD_L3SGSSTHA1100100.h5.gz
```

ダウンロードしたファイルは、gzip で圧縮された HDF5 というファイルフォーマットの数値データです。HDF5 は科学技術用に開発された数値データを格納するためのバイナリーフォーマットで多次元の数値データだけでなく、観測日時や観測者、スケーリングファクタ、解析アルゴリズムといった情報を階層構造で格納することができるフォーマットです。もちろん仕様は公開されており Python や Ruby のモジュールもあります。

この中に海面温度の数値データが北緯 0 度、東経 0 度から 0.1 度刻みのメッシュで格納されています。まずこのメッシュ 1 つを 1px とする画像を作ります。そうすると地球全体で 3600px×1800px の画像ができます。

私は Python を使って、メッシュのデータを取り出し、あらかじめ作っておいたカラーマップで数値を RGB に変換し、ASCII テキストでピクセルを表現できる PPM で出力しました。欠損値は (R,G,B)=(0,0,0) としています。

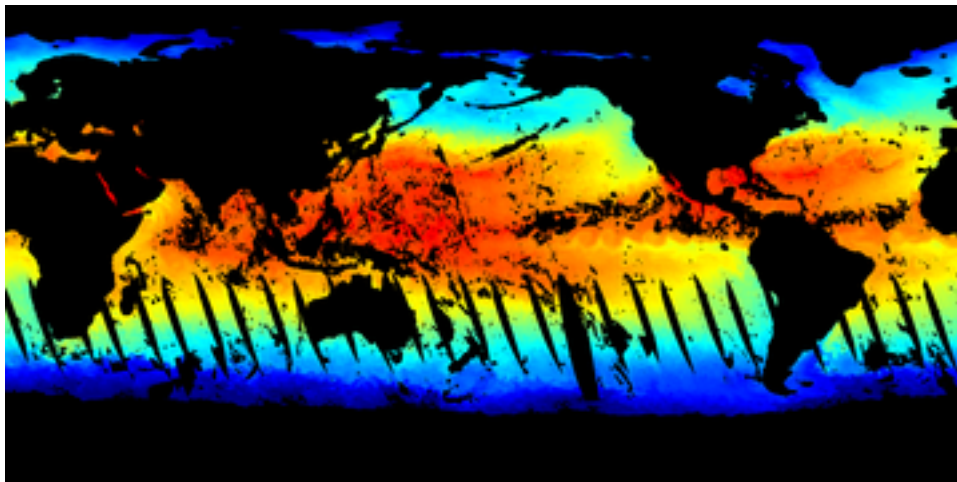


図 2 生成したマップ

この画像を ImageMagick の convert コマンドで tiff 画像にしつつ、欠損値を表す黒を透明に変換します。

```
$ convert -transparent black converted.ppm map.tiff
```

ここでできる Tiff 画像 (図 2) は、地図ではありますが、ピクセルと緯度経度の対応関係の情報を持っていません。そこで GDAL を使って、この Tiff 画像を位置情報を持つ GeoTiff というフォーマットに変換します。GDAL は、地理空間データのスイスアーミーナイフ的な存在で、GeoTiff や JPG、PNG といったラスターデータフォーマットや HDF、NetCDF 等のファイルを読んで他のフォーマットに変換したり、測地系の変換等を行うことができるライブラリ、ユーティリティ群です。(ここまででやってきた、HDF5 ファイルから数値データを取り出してカラーマップにすることも、GDAL だけでできる気がします。) GDAL は、gdal-bin というパッケージになっています。Tiff ファイルを GeoTiff に変換するには、gdal_translate コマンドで、緯度経度が分かっているピクセルを教えます。そして gdalwarp コマンドで、gdal_translate で指定したピクセルと緯度経度を元に幾何変換して測地系の情報を追加します。


```
$ gdal_translate -q -gcp 0 0 0 90 \  
-gcp 3600 0 360 90 \  
-gcp 0 1800 0 -90 \  
-gcp 3600 1800 360 -90 map.tiff tmp.tiff  
  
$ gdalwarp -q -s_srs EPSG:4326 -r cubic tmp.tiff map.tiff
```

各ピクセルが位置情報を持った画像ファイルが完成しました！これでようやくタイルを生成する準備ができました。タイルの生成には、python-gdal パッケージの `gdal2tiles.py` コマンドを使います。ただこのコマンドは、指定したズームレベルのタイルをすべて生成してしまいます。前述の通り、ズームレベルを増やすと指数関数的に生成する画像が増えていくのでタイルの生成に数時間かかってしまいます。数時間かけて苦労してタイルを作っても、現実にはユーザーがすべてのタイルを見てくれるわけではありません。だったら必要なタイルだけを、必要な時に生成すれば良いのです。

4.3 タイルマップサーバの実装

では、リクエストがあったときに必要に応じてタイルを生成するサーバを作りましょう。今回は Web サーバは Apache で、`mod_python` と `mod_rewrite` を用います。`mod_rewrite` の説明は不要でしょう。正規表現にマッチしたリクエストの URL を書き換えるモジュールです。

`mod_python` は、Apache の Python binding で、CGI の置き換えを目的としたモジュールです。インストールするには、以下のコマンドを実行します。

```
# apt-get install libapache2-mod-python  
# a2enmod python  
# service apache2 restart
```

設定は、`/etc/conf.d/` の中に適当なファイル名のファイルを作って

```
<Directory /some/path>  
  SetHandler mod_python  
  PythonHandler mod_python.publisher  
</Directory>
```

としればよいでしょう。

`mod_python` の `mod_python.publisher` は、HTTP リクエストを Python の関数の呼び出しに変換します。たとえば、`www.example.org` の DocumentRoot に、`hello.py` という以下の Python スクリプトがあったとしましょう。

```
def sayHello(req, name):  
    return 'Hello %s'%name
```

もしも、`http://www.example.org/hello.py/sayHello?name=Debian` に GET リクエストがあると、Web サーバは、`sayHello` 関数の `name` 引数に `Debian` を代入して `sayHello` 関数を呼び出し、“Hello Debian” と表示されます。

タイルマップサービスにアクセスするクライアントは、`http://BASEURL/VERSION/TILENAME/z/x/y.png` という URI で GET しようとするので、`mod_rewrite` で URI を書き換えましょう。たとえば、`www.example.org` の `/etc/apache/conf.d` に以下のように設定すると、

```
<Directory "/var/www/tile/1.0.0/sst">  
  RewriteEngine On  
  RewriteBase /tile/1.0.0/sst/  
  RewriteRule ^([0-9]+)/([0-9]+)/([0-9]+).png gettile.py/get?z=$1&x=$2&y=$3  
  
  AddHandler mod_python .py  
  PythonHandler mod_python.publisher  
</Directory>
```

`http://www.example.org/tile/1.0.0/sst/4/12/34.png` が、`http://www.example.org/tile/1.0.0/sst/gettile.py/get?z=4&x=12&y=34` に変換されるので、`gettile.py` の中の、`req`, `x`, `z`, `y` の引数をとる関数が呼び出されます。ここに指定された URI のタイルだけを生成処理を書けば、オンデマンドにタイルを生成するタイルマップサービスができます。

単一タイルだけを生成するには、gdal2tiles.py を少し修正します。generate_base_tiles 関数の中の 1180 行目付近にすべてのタイルを作っているループが存在するので、

```
def generate_base_tiles(self):
    """Generation of the base tiles (the lowest in the pyramid) directly from the input raster"""
    (Snip...)

    for ty in range(tmaxy, tminy-1, -1): #range(tminy, tmaxy+1):
        for tx in range(tminx, tmaxx+1):

            if self.stopped:
                break
            ti += 1
            tilefilename = os.path.join(self.output, str(tz), str(tx), "%s.%s" % (ty, self.tileext))
            if self.options.verbose:
                print(ti, '/', tcount, tilefilename) #, "( TileMapService      : z / x / y )"
```

それをひも解いて指定されたタイルを生成している部分を、関数として抜き出します。

```
def generate_base_tiles(self):
    """Generation of the base tiles (the lowest in the pyramid) directly from the input raster"""
    (Snip...)

    for ty in range(tmaxy, tminy-1, -1): #range(tminy, tmaxy+1):
        for tx in range(tminx, tmaxx+1):
            generate_tile(tx, ty, self.tmaxz)

# -----
def generate_tile(self, ty, tx, tz):
    ds = self.out_ds
    tilebands = self.dataBandsCount + 1
    querysize = self.querysize
```

あとは、generate_tile 関数を、mod_python が呼ぶスクリプトから呼べばできあがりです。

```
def get(req, z, tx, ty):
    req.content_type = 'image/png'
    #req.write('z: %s\ntx: %s\nty: %s'%(z,tx,ty))

    g = GDAL2Tiles(['/home/chome/public_html/tile/sst/map.tiff', '/var/www/tile/1.0.0/sst'])
    g.open_input()
    g.generate_tile(int(ty), int(tx), int(z))

    with open('/var/www/tile/1.0.0/sst/%s/%s/%s.png'%(z,tx,ty), 'rb') as f:
        req.write(f.read())
```

4.4 まとめ

Debian でパッケージングされているオープンソースのプログラムやライブラリを組み合わせでタイルマップサービスを作ってみました。gdal2tiles.py を少し修正し mod_python から呼ぶことによって、オンデマンドで必要なタイルだけを生成しています。

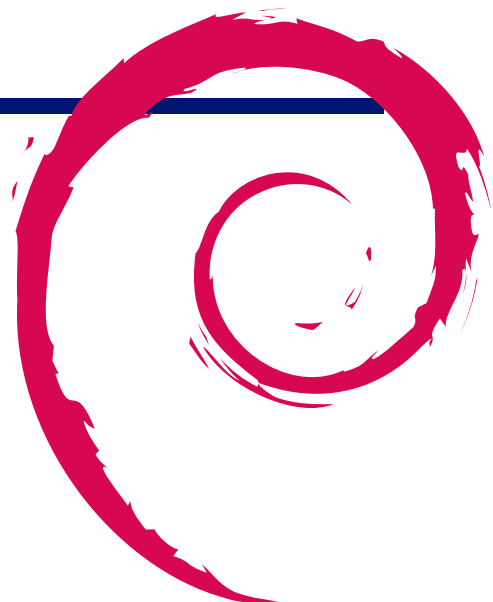
今回開発したサービスは <http://www.hi-rezclimate.org> で公開されています。データの取得、GeoTiff にするところまでは自動化されているので、毎日最新のデータをダウンロードして自動で更新されています。「しずく」は海水面温度だけでなく、海上風速や水蒸気量、積雪深といった水に関する現象を観測しています。今後サポートするデータを増やしていく予定です。

参考文献

- [1] “Tile Map Service in Geocode”, http://geoikia.idgis.eu/wiki-english/index.php/Tile_Map_Services_in_Geocode
- [2] “Slippy map tilenames”, http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames
- [3] GCOM-W1 データ提供サービス <https://gcom-w1.jaxa.jp/auth.html>

5 索引

debian-tilemap-service, 6





Debian 勉強会資料

2014年08月23日 初版第1刷発行

東京エリア Debian 勉強会（編集・印刷・発行）
